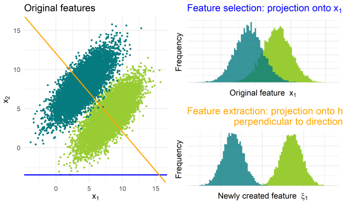
Supervised Learning

Feature Selection



Learning goals

- Understand that adding more features can be detrimental to prediction performance.
- Understand the benefits of keeping only informative features for the model.

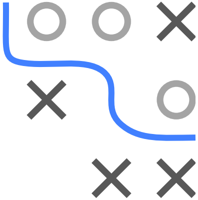
INTRODUCTION

Feature selection deals with

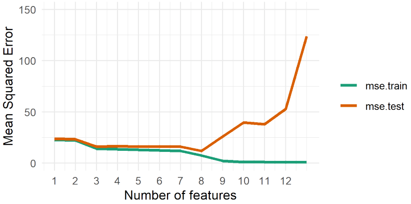
- techniques for choosing a suitable subset of features
- evaluating the influence of features on the model

Feature selection can be performed relying on domain knowledge and expert input, or using a data-driven algorithmic approach.

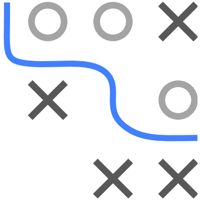
MOTIVATION



- Naive view:
  - More features $\rightarrow$ more information $\rightarrow$ discriminant power $\uparrow$
  - Model is not harmed by irrelevant features since their parameters can simply be estimated as 0.
- In practice, irrelevant and redundant features can “confuse” learners (see **curse of dimensionality**) and worsen performance.
- Example: In linear regression, R^2 is monotonically increasing in p, but adding irrelevant features leads to overfitting (capturing noise).



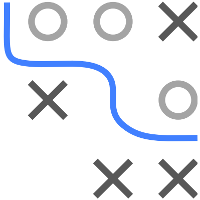
MOTIVATION



- In high-dimensional data sets, we often have prior information that many features are either irrelevant or redundant.
- Feature selection is critical for
  - reducing noise and overfitting,
  - improving performance/generalization,
  - interpretability by identifying most informative features.
- Feature selection can also remedy problems arising in small n regimes or under limited computational resources.
- Many models require $n > p$ data. Thus, we either need to
  - adapt models to high-dimensional data (e.g. regularization),
  - design entirely new procedures for $p > n$ data, or
  - use the preprocessing methods addressed in this lecture.

SIZE OF DATASETS

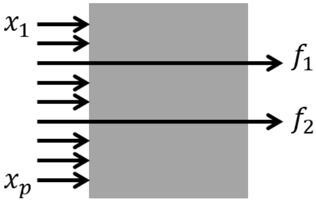
The increasingly automatized collection of information makes data sets with extremely high dimensionality available, while classical models were developed for small p data.



- **Classical setting**: Up to around 10^2 features, feature selection might be relevant, but benefits often negligible.
- **Datasets of medium to high dimensionality**: At around 10^2 to 10^3 features, classical approaches can still work well, while principled feature selection helps in many cases.
- **High-dimensional data**: 10^3 to 10^9 or more features. Examples are e.g. micro-array / gene expression data and text categorization (bag-of-words features). If, in addition, observations are few, the scenario is called $p \gg n$.

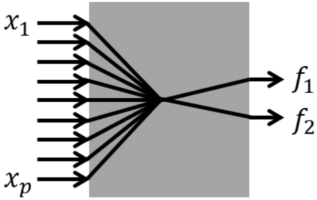
FEATURE SELECTION VS. EXTRACTION

Feature selection



- Creates a subset of original features $\mathbf{x}$ by selecting $\tilde{p} < p$ features $\boldsymbol{f}$.
- Retains information on selected individual features.

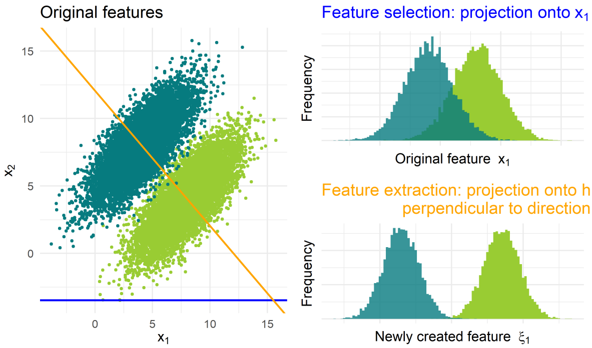
Feature extraction



- Maps p features in $\mathbf{x}$ to $\tilde{p}$ extracted features $\boldsymbol{f}$.
- Info on individual features can be lost through (non-)linear combination.

FEATURE SELECTION VS. EXTRACTION

- Both FS and FE contribute to
  1) dimensionality reduction, and 2) simplicity of classification rules.
- FE can be unsupervised (PCA, Multidimensional Scaling, Manifold Learning) or supervised (supervised PCA, partial least squares).
- FE can produce lower dim projections which can be more informative than FS.



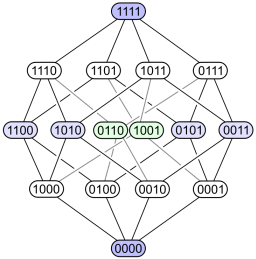
OBJECTIVE FUNCTION

Given p features, the **best-subset selection problem** is to find a subset $S \subseteq \{1, \dots p\}$ optimizing objective $\Psi : \Omega \to \mathbb{R}$:

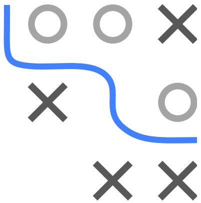
$$S^* \in \arg\min_{S \in \Omega} \{\Psi(S)\}$$

- Ω = search space of all feature subsets $S \subseteq \{1, \dots, p\}$. Usually we encode this by bit vectors, i.e., $\Omega = \{0, 1\}^p$ (1 = feat. selected)

- Objective Ψ can be different functions, e.g., AIC/BIC for LM or cross-validated performance of a learner.



HOW DIFFICULT IS BEST-SUBSET SELECTION?



- Size of search space = 2^p, i.e., grows exponentially in p as it is the power set of $\{1, \ldots, p\}$.
- Finding best subset is discrete combinatorial optimization problem also known as L_0 regularization.
- It can be shown that this problem unfortunately can not be solved efficiently in general (NP-hard; see, e.g., Natarajan, 1995).
- We can avoid having to search the entire space by employing efficient search strategies, moving through the search space in a smart way that finds performant feature subsets.